

Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications^{*}

Jeroen Delvaux^{1,2}, Dawu Gu², Ingrid Verbauwhede¹,
Matthias Hiller³ and Meng-Day (Mandel) Yu^{4,1,5}

¹ KU Leuven, ESAT/COSIC and iMinds,
Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
{jeroen.delvaux, ingrid.verbauwhede}@esat.kuleuven.be

² Shanghai Jiao Tong University, CSE/LoCCS,
800 Dongchuan Road, Shanghai 200240, China
dwgu@sjtu.edu.cn

³ Chair of Security in Information Technology,
Technical University of Munich, Germany
matthias.hiller@tum.de

⁴ Verayo Inc., USA
myu@verayo.com

⁵ CSAIL, MIT, USA

Abstract. The device-unique response of a *physically unclonable function* (PUF) can serve as the root of trust in an embedded cryptographic system. *Fuzzy extractors* transform this noisy non-uniformly distributed secret into a stable high-entropy key. The overall efficiency thereof, typically depending on error-correction with a binary $[n, k, d]$ block code, is determined by the universal and well-known $(n - k)$ bound on the min-entropy loss. We derive new considerably tighter bounds for PUF-induced distributions that suffer from, e.g., bias or spatial correlations. The bounds are easy-to-evaluate and apply to large non-trivial codes, e.g., BCH, Hamming and Reed-Muller codes. Apart from an inherent reduction in implementation footprint, the newly developed theory also facilitates the analysis of state-of-the-art error-correction methods for PUFs. As such, we debunk the reusability claim of the reverse fuzzy extractor. Moreover, we provide proper quantitative motivation for debiasing schemes, as this was missing in the original proposals.

Keywords: fuzzy extractor, secure sketch, min-entropy, physically unclonable function, coding theory

1 Introduction

Cryptography relies on reproducible uniformly distributed secret keys. Obtaining affordable physically secure key-storage in embedded non-volatile memory is

^{*} This manuscript is accepted for the CHES 2016 conference. An extended version is available in the Cryptology ePrint Archive, Report 2015/854.

hard though. Harvesting entropy from *physically unclonable functions* (PUFs) comprehends an alternative that lowers the vulnerability during the power-off state. Unfortunately, PUF responses are corrupted by noise and non-uniformities are bound to occur. A *fuzzy extractor* [11] provides an *information-theoretically secure* mechanism to convert PUF responses into high-quality keys. The essential building block for handling noisiness is the *secure sketch*, providing error-correction with most frequently a binary $[n, k, d]$ block code. Associated public helper data reveals information about the PUF response though; the system provider should hence quantify how much min-entropy remains. So far, the conservative $(n - k)$ upper bound on the min-entropy loss has been applied. Unfortunately, the residual min-entropy is underestimated, implying that more PUF response bits than necessary have to be used. Expensive die area is hence blocked by PUF circuits that are not strictly required to obtain the desired security level, i.e., symmetric key length.

1.1 Contribution

The novelty of our work is twofold:

- First, we derive new bounds on the secure sketch min-entropy loss for PUF-induced distributions with practical relevance. Our bounds are considerably tighter than the well-known $(n - k)$ formula, hereby improving the implementation efficiency of PUF-based key generators. The discrepancy is showcased for two predominant PUF imperfections, i.e., biased and spatially correlated response bits. It is important to note that a variety of commonly used codes is covered, e.g., BCH and Reed-Muller codes, regardless of their algebraic complexity. Furthermore, a large variety of distributions could be supported. Therefore, our scope reaches considerably further than related work in [8, 22], focussing on simple repetition codes and biased distributions only. As in the latter works, our bounds are easy-to-evaluate and able to support large codes.
- Second, the newly developed theory is applied to state-of-the-art error-correction methods for PUFs. As such, we reveal a fundamental flaw in the reverse fuzzy extractor, proposed by Van Herrewege et al. [28] at Financial Crypto 2012. The latter lightweight primitive is gaining momentum and has also been adopted in the CHES 2015 protocol of Aysu et al. [1]. We debunk the main security claim that repeated helper data exposure does not result in additional min-entropy loss. Furthermore, we contribute to the motivation of debiasing schemes such as the *index-based syndrome* (IBS) proposal of Yu et al. [30], and the CHES 2015 proposal of Maes et al. [22]. The latter proposals assume that a stand-alone sketch cannot handle biased distributions. We eliminate the need for an educated guess that originates from the extrapolation of repetition code insights and/or the application of the overly conservative $(n - k)$ bound.

1.2 Organization

The remainder of this manuscript is organized as follows. Section 2 introduces notation and preliminaries. Section 3 derives new tight bounds on the secure sketch min-entropy loss. Section 4 elaborates applications of the newly developed theory. Section 5 concludes the work.

2 Preliminaries

2.1 Notation

Binary vectors are denoted with a bold lowercase character, e.g., \mathbf{x} . All vectors are row vectors. All-zeros and all-ones vectors are denoted with $\mathbf{0}$ and $\mathbf{1}$ respectively. Binary matrices are denoted with a bold uppercase character, e.g., \mathbf{H} . A random variable and its corresponding set of outcomes are denoted with an uppercase *italic* and calligraphic character respectively, e.g., X and \mathcal{X} . Variable assignment is denoted with an arrow, e.g., $\mathbf{x} \leftarrow X$. Custom-defined procedure names are printed in a sans-serif font, e.g., Hamming weight $\text{HW}(\mathbf{x})$ and Hamming distance $\text{HD}(\mathbf{x}, \tilde{\mathbf{x}})$. The probability of an event A is denoted as $\mathbb{P}(A)$. The expected value of a function $g(X)$ of random variable X is denoted as $\mathbb{E}_{x \leftarrow X}[g(X)]$. The probability density function and cumulative distribution function of a standard normal distribution $N(0, 1)$ are denoted as $\mathbf{f}_{\text{norm}}(\cdot)$ and $\mathbf{F}_{\text{norm}}(\cdot)$ respectively. For a binomial distribution with n trials and success probability p , we use $\mathbf{f}_{\text{bino}}(\cdot; n, p)$ and $\mathbf{F}_{\text{bino}}(\cdot; n, p)$ respectively.

2.2 Min-Entropy Definitions

The *min-entropy* of a random variable X is as defined in (1). Consider now a pair of possibly correlated random variables: X and P . The *conditional min-entropy* [11] of X given P is as defined in (2). Terms with $\mathbb{P}(P = p) = 0$ are evaluated as 0. Both definitions quantify the probability that an attacker guesses $x \leftarrow X$ first time right, on a logarithmic scale. We emphasize that min-entropy is a more conservative notion than Shannon entropy and therefore often preferred within cryptography.

$$\mathbb{H}_{\infty}(X) = -\log_2\left(\max_{x \in \mathcal{X}} \mathbb{P}(X = x)\right). \quad (1)$$

$$\tilde{\mathbb{H}}_{\infty}(X|P) = -\log_2\left(\mathbb{E}_{p \leftarrow P}\left[\max_{x \in \mathcal{X}} \mathbb{P}((X = x)|(P = p))\right]\right). \quad (2)$$

2.3 Physically Unclonable Functions

A prominent category of PUFs, suitable for key generation in particular, consists of an array of identically designed cells. Each cell produces a single bit, or occasionally a few bits. This includes memory-based designs, such as the SRAM PUF [16], as well as the coating PUF [25] and a subset of the large number of

ring oscillator-based designs, e.g., [29]. The most prominent entropy-degrading effects for such PUFs are bias and spatial correlations. Bias comprehends an imbalance between the number of zeros and ones. Spatial correlations implicate that neighboring cells might influence each other.

We describe a parameterized probability distribution for the error rate of individual PUF response bits $\tilde{\mathbf{x}}(i)$, with $i \in [1, n]$. Experimental validation on various PUF circuits, e.g., in [20, 10], labelled the model as accurate. Two hidden random variables are incorporated: the normalized manufacturing variability $V_i \sim N(0, 1)$, drawn only once for each response bit, and additive noise $N_{ij} \sim N(0, \sigma_N)$, drawn for each evaluation j of a given response bit. A response bit $\tilde{\mathbf{x}}(i)$ evaluates to 1 if $(v_i + n_{ij}) > t$ and 0 otherwise, with threshold t a fixed parameter. Bias corresponds to a nonzero t . Spatial correlations can be incorporated via a multivariate normal distribution $(V_1 \dots V_n) \sim N(\mathbf{0}, \mathbf{\Sigma})$, with $\mathbf{\Sigma}$ the symmetric $n \times n$ covariance matrix.

For ease of analysis, we consider the response bits $\mathbf{x}(i)$ obtained by thresholding $v_i > t$ as a reference. In practice, these nominal values can be approximated via a majority vote among noisy replicas $\tilde{\mathbf{x}}(i)$, possibly accelerated via circuit techniques [4, 30]. Bias parameter b , defined as the probability $\mathbb{P}(\mathbf{x}(i) = 1)$, then equals $F_{\text{norm}}(-t)$. Zero bias corresponds to $b = 0.5$. The error rate p_E of a response bit $\tilde{\mathbf{x}}(i)$ with respect to its reference, i.e., the probability $\mathbb{P}(\mathbf{x}(i) \neq \tilde{\mathbf{x}}(i))$, then equals $F_{\text{norm}}(-|v_i - t|/\sigma_N)$.

2.4 Secure Sketch and Fuzzy Extractor Definitions

Secure sketches operate on a metric space \mathcal{X} with distance function dist . For PUFs, we can restrict our attention to binary vectors $\mathbf{x} \in \{0, 1\}^{1 \times n}$ and the Hamming distance HD therebetween. An attacker knows the probability distribution of $\mathbf{x} \leftarrow X$. Consider a noisy version $\tilde{\mathbf{x}}$ of sample \mathbf{x} . A *secure sketch* [11] is a pair of efficient and possibly randomized procedures: the sketching procedure $\mathbf{p} \leftarrow \text{SSGen}(\mathbf{x})$, with helper data $\mathbf{p} \in \mathcal{P}$, and the recovery procedure $\hat{\mathbf{x}} \leftarrow \text{SSRep}(\tilde{\mathbf{x}}, \mathbf{p})$. There are two defining properties:

- *Correctness*. If $\text{HD}(\mathbf{x}, \tilde{\mathbf{x}}) \leq t$, correctness of reconstruction is guaranteed, i.e., $\hat{\mathbf{x}} = \mathbf{x}$. If $\text{HD}(\mathbf{x}, \tilde{\mathbf{x}}) > t$, there is no guarantee whatsoever.
- *Security*. Given a certain lower bound h_{in} on the ingoing min-entropy, i.e., $\mathbb{H}_{\infty}(X) \geq h_{\text{in}}$, a corresponding lower bound h_{out} on the residual min-entropy, i.e., $\mathbb{H}_{\infty}(X|\mathbf{p}) \geq h_{\text{out}}$, can be imposed. Often, but not necessarily, this condition can be satisfied regardless of h_{in} . Or stated otherwise, there is a certain upper bound on the min-entropy loss $\Delta\mathbb{H}_{\infty} = \mathbb{H}_{\infty}(X) - \mathbb{H}_{\infty}(X|\mathbf{p})$.

A slightly modified notion brings us to the *fuzzy extractor* [11]. Output $\mathbf{k} \in \mathcal{K}$ is then required to be nearly-uniform, given observation $\mathbf{p} \leftarrow P$, and is therefore suitable as a secret key. There is a proven standard method to craft a fuzzy extractor from a secure sketch. In particular, a *randomness extractor* could derive a key from the secure sketch output, i.e., $\mathbf{k} \leftarrow \text{Ext}(\mathbf{x})$. *Universal hash functions* [7] are good randomness extractors, according to the (*generalized*) *leftover*

hash lemma [13, 2]. Unfortunately, their min-entropy loss is quite substantial. In practice, key generators therefore often rely on a cryptographic hash function that is assumed to behave as a *random oracle*. The latter idealized heuristic results in zero min-entropy loss.

2.5 Coding Theory

A *binary code* \mathcal{C} is a bijection from a message space \mathcal{M} to a codeword space $\mathcal{W} \subseteq \{0, 1\}^{1 \times n}$. The *minimum distance* d is the minimum number of bits in which any two distinct codewords differ. A procedure $\mathbf{w} \leftarrow \text{Encode}(\mathbf{m})$ maps a message $\mathbf{m} \in \mathcal{M}$ to a codeword $\mathbf{w} \in \mathcal{W}$. A procedure $\hat{\mathbf{w}} \leftarrow \text{Correct}(\tilde{\mathbf{w}})$ corrects up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors for any noise-corrupted codeword $\tilde{\mathbf{w}} = \mathbf{w} \oplus \mathbf{e}$, with $\text{HW}(\mathbf{e}) \leq t$. Equation (3) expresses the Hamming bound [18]. The equality holds for *perfect codes* only, implicating that any vector in $\{0, 1\}^{1 \times n}$ is within distance t of a codeword. All other codes are subject to the inequality.

$$\sum_{i=0}^t \binom{n}{i} |\mathcal{M}| \leq 2^n. \quad (3)$$

A binary $[n, k, d]$ *block code* \mathcal{C} restricts the message length $k = \log_2(|\mathcal{M}|)$ to an integer. For a linear block code, any linear combination of codewords is again a codeword. A $k \times n$ *generator matrix* \mathbf{G} , having full rank, can then implement the encoding procedure, i.e., $\mathbf{w} = \mathbf{m} \cdot \mathbf{G}$. For any translation $\boldsymbol{\tau} \in \{0, 1\}^{1 \times n}$ and linear code \mathcal{C} , the set $\{\boldsymbol{\tau} \oplus \mathbf{w} : \mathbf{w} \in \mathcal{W}\}$ is referred to as a *coset*. Two cosets are either disjoint or coincide. Therefore, the vector space $\{0, 1\}^{1 \times n}$ is fully covered by 2^{n-k} cosets, referred to as the *standard array*. The minimum weight vector $\boldsymbol{\epsilon}$ in a coset is called the *coset leader*. In case of conflict, i.e., a common minimum $\text{HW}(\boldsymbol{\epsilon}) > t$, an arbitrary leader can be selected. The minimum distance d of a linear code equals the minimum Hamming weight of its nonzero codewords. A linear code \mathcal{C} is *cyclic* if every circular shift of a codeword is again a codeword belonging to \mathcal{C} .

2.6 The Code-Offset Secure Sketch

Several secure sketch constructions rely on a binary code \mathcal{C} . For ease of understanding, we focus on the code-offset method of Dodis et al. [11] exclusively. Nevertheless, equivalencies in the extended version of this manuscript (Cryptology ePrint Archive, Report 2015/854) prove that all results apply to six other constructions equally well. The code \mathcal{C} that instantiates the code-offset method in Fig. 1 is not necessarily linear. Even more, it is not required be a block code either. Linear codes (BCH, Hamming, repetition, etc.) remain the most frequently used though due to their efficient decoding algorithms [18]. Correctness of reconstruction is guaranteed if $\text{HD}(\mathbf{x}, \tilde{\mathbf{x}}) \leq t$, with t the error-correcting capability of the code.

$p \leftarrow \text{SSGen}(x)$	$\hat{x} \leftarrow \text{SSRep}(\tilde{x}, p)$
Random $w \in \mathcal{C}$	$\tilde{w} \leftarrow \tilde{x} \oplus p = w \oplus e$
$p \leftarrow x \oplus w$	$\hat{x} \leftarrow p \oplus \text{Correct}(\tilde{w})$

Fig. 1. The code-offset secure sketch, having an n -bit reference input x .

Min-entropy loss can be understood as a *one-time pad* imperfection. Sketch input x is masked with a random codeword w , i.e., an inherent entropy deficiency: $\mathbb{H}_\infty(W) = \log_2(|\mathcal{M}|) < n$. For linear codes in particular, we highlight a convenient interpretation using cosets. Helper data p then reveals in which coset reference x resides. It can be seen easily that p is equal to a random vector in the same coset as x . The residual min-entropy in (2) hence reduces to (4) for linear codes, with e a coset leader. We emphasize that the min-entropy loss $\Delta\mathbb{H}_\infty$ does not depend on the decoding method, simply because the helper data is not affected. For $[n, k, d]$ block codes in particular, the well-known upper bound $\Delta\mathbb{H}_\infty \leq (n - k)$ holds, as proven in [11]. More generally, this extends to $\Delta\mathbb{H}_\infty \leq n - \log_2(|\mathcal{M}|)$.

$$\tilde{\mathbb{H}}_\infty(X|P) = -\log_2\left(\mathbb{E}_{e \leftarrow E}\left[\max_{w \in \mathcal{W}} \mathbb{P}((X = e \oplus w)|(E = e))\right]\right). \quad (4)$$

3 Tight Bounds on the Min-Entropy Loss

Currently, secure sketch implementations rely on the $(n - k)$ upper bound on the min-entropy loss, e.g., [23]. Unfortunately, this leads to an overly conservative design when instantiating security parameters accordingly. We develop a graphical framework that produces tight bounds on $\tilde{\mathbb{H}}_\infty(X|P)$ for typical PUF-induced distributions. The critical *first-order* effects of bias and spatial correlations are captured. Both lower and upper bounds are supported. The lower bounds are of primary interest for a conservative system provider, entertaining the worst-case scenario. We considerably improve upon the $(n - k)$ bound, i.e., the leftmost inequality in (5). We also improve upon the rather trivial upper bounds [11] that comprehend the rightmost inequality in (5).

$$\underbrace{\max(\mathbb{H}_\infty(X) - (n - \log_2(|\mathcal{M}|)), 0)}_{\text{worst-case}} \leq \tilde{\mathbb{H}}_\infty(X|P) \leq \underbrace{\min(\log_2(|\mathcal{M}|), \mathbb{H}_\infty(X))}_{\text{best-case}}. \quad (5)$$

Our lower and upper bounds combined define a relatively narrow interval in which the exact value of $\tilde{\mathbb{H}}_\infty(X|P)$ is enclosed. We considerably extend related work in [8, 22] as follows. First, we cover a variety of codes, regardless of their algebraic complexity. Prior work focussed on repetition codes only. Although frequently used as the inner code of a concatenated code [5], full-fledged key generators [23] typically rely on non-trivial codes, e.g., BCH codes [18]. Second,

our techniques may be applied to a variety of distributions, while prior work covered biased distributions only. Our bounds remain easy-to-evaluate and are able to handle large codes. Although derived for the code-offset sketch of Dodis et al. [11] in particular, the extended version of this manuscript establishes the equivalence with six other constructions.

3.1 Distributions

Our work is generic in the sense that a large variety of distributions X could be covered. We only require that $\mathcal{X} = \{0, 1\}^{1 \times n}$ can be partitioned in a limited number of subsets φ_j , with $j \in [1, J]$, so that all elements of φ_j have the same probability of occurrence q_j . Formally, $\mathbb{P}(X = \mathbf{x}) = q_j$ if and only if $\mathbf{x} \in \varphi_j$. These probabilities are strictly monotonically decreasing, i.e., $q_1 > q_2 > \dots > q_J$. Occasionally, $q_J = 0$. The ingoing min-entropy is easily computed as $\mathbb{H}_\infty(X) = -\log_2(q_1)$.

We determine bounds on $\tilde{\mathbb{H}}_\infty(X|P)$. The runtime of the corresponding algorithms is roughly proportional to J . The crucial observation is that even a very small J might suffice to capture realistic PUF models. Below, we describe a parameterized distribution X for both biased and spatially correlated PUFs. Both distributions are to be considered as proof-of-concept models, used in showcasing the feasibility of a new research direction. In case a given PUF is not approximated accurately enough, one can opt for an alternative and possibly more complicated *second-order* distribution. As long as J is limited, bounds can be evaluated in milliseconds-minutes on a standard desktop computer.

- *Biased distribution.* We assume response bits to be independent and identically distributed (i.i.d.) so that $\mathbb{P}(X(i) = 1) = b$, with $i \in [1, n]$ and a real-valued $b \in [0, 1]$. For $b = \frac{1}{2}$, this corresponds to a uniform distribution. The latter bias model comprehends a very popular abstraction in PUF literature. The min-entropy loss of various other helper data methods has been analyzed as such, e.g., *soft-decision decoding* [21, 8] as well as *IBS* [30, 15] and von Neumann [22, 27] debiasing. Therefore, our results enable adequate comparison with related methods, all using a common baseline distribution.
- *Correlated distribution.* We assume response bits to be distributed so that $\mathbb{P}(X(i) = X(i+1)) = c$, with $i \in [1, n-1]$ and a real-valued $c \in [0, 1]$. This extends to (6) for larger neighborhoods. There is no bias, i.e., $\mathbb{P}(X(i) = 1) = \frac{1}{2}$. For $c = \frac{1}{2}$, the latter model corresponds to a uniform distribution. Although spatial correlations are frequently encountered in experimental work, e.g., byte-level dependencies for the SRAM PUFs in [14, 1], these are often neglected in information theoretic work due to their complexity. We hope that our results may help turn the tide on this.

$$\mathbb{P}(X(i) = X(j)) = \sum_{u=0}^{\lfloor |i-j|/2 \rfloor} \text{f}_{\text{bino}}(2u; |i-j|, 1-c), \quad \text{with } i, j \in [1, n]. \quad (6)$$

Fig. 2 specifies the subsets φ_j for both distributions. For the biased distribution, we partition according to $\text{HW}(\mathbf{x})$. This corresponds to a binomial distribution with $j - 1$ successes for n Bernoulli trials, each having success probability $b_\star = \min(b, 1 - b)$. For the correlated distribution, we partition according to $\text{HD}(\mathbf{x}(1 : n - 1), \mathbf{x}(2 : n))$, i.e., the number of transitions in \mathbf{x} . Inputs in subset φ_j exhibit $j - 1$ transitions and obey either one out of two forms, i.e., $\mathbf{x} = (\mathbf{0}\|\mathbf{1}\|\mathbf{0}\|\dots)$ and $\mathbf{x} = (\mathbf{1}\|\mathbf{0}\|\mathbf{1}\|\dots)$. A related observation is that if $\mathbf{x} \in \varphi_j$, then so is its ones' complement, i.e., $\bar{\mathbf{x}} \in \varphi_j$. This explains the factors 2 and $\frac{1}{2}$ everywhere. Set size $|\varphi_j|$ is further determined with *stars and bars* combinatorics [12]. In particular, we separate n indistinguishable stars into j distinguishable bins by adding $j - 1$ out of $n - 1$ bars.

j	$ \varphi_j $	q_j	j	$ \varphi_j $	q_j
1	1	$(1 - b_\star)^n$	1	2	$\frac{1}{2}(1 - c_\star)^{n-1}$
2	n	$b_\star(1 - b_\star)^{n-1}$	2	$2(n - 1)$	$\frac{1}{2}c_\star(1 - c_\star)^{n-2}$
...
j	$\binom{n}{j-1}$	$(b_\star)^{j-1}(1 - b_\star)^{n-j+1}$	j	$2\binom{n-1}{j-1}$	$\frac{1}{2}(c_\star)^{j-1}(1 - c_\star)^{n-j}$
...
n	n	$(b_\star)^{n-1}(1 - b_\star)$	$n - 1$	$2(n - 1)$	$\frac{1}{2}(c_\star)^{n-2}(1 - c_\star)$
$n + 1$	1	$(b_\star)^n$	n	2	$\frac{1}{2}(c_\star)^{n-1}$

Fig. 2. Subsets φ_j for a biased and correlated distribution X , left and right respectively. We define $b_\star = \min(b, 1 - b)$ and $c_\star = \min(c, 1 - c)$.

We treat the degenerate case $b = c = \frac{1}{2}$, i.e., a uniform distribution, separately. There is only one set then. Formally, $J = 1$, $|\varphi_1| = 2^n$ and $q_1 = 1/2^n$. As proven by Reyzin [24], the min-entropy loss of a secure sketch is maximal for a uniformly distributed input, making this a case of special interest.

3.2 Generic Bounds

Equation (7) holds for the code-offset construction of Dodis et al. [11], given that a codeword is selected fully at random during enrollment.

$$\mathbb{P}((P = \mathbf{p})|(X = \mathbf{x})) = \begin{cases} 1/|\mathcal{M}|, & \text{if } \exists \mathbf{w} : \mathbf{p} = \mathbf{x} \oplus \mathbf{w} \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

Equation (8) applies Bayes' rule to the definition of conditional min-entropy in (2) and fills in (7). The 0 case is resolved by switching variables for the max operator. A direct exhaustive evaluation of the resulting formula requires up to $2^n|\mathcal{M}|$ operations.

$$\begin{aligned}
\tilde{\mathbb{H}}_\infty(X|P) &= -\log_2 \left(\sum_{\mathbf{p} \in \mathcal{P}} \frac{\mathbb{P}(P=\mathbf{p})}{\mathbb{P}(P=\mathbf{p})} \max_{\mathbf{x} \in \mathcal{X}} \frac{\mathbb{P}(X=\mathbf{x})\mathbb{P}((P=\mathbf{p})|(X=\mathbf{x}))}{\mathbb{P}(P=\mathbf{p})} \right) \\
&= -\log_2 \left(\frac{1}{|\mathcal{M}|} \sum_{\mathbf{p} \in \mathcal{P}} \max_{\mathbf{w} \in \mathcal{W}} \mathbb{P}(X=\mathbf{p} \oplus \mathbf{w}) \right). \tag{8}
\end{aligned}$$

For linear codes, the workload can be reduced substantially. With a similar derivation as before, we rewrite (4) as shown in (9). Up to 2^n operations suffice. Nevertheless, direct evaluation is only feasible for small codes. We emphasize that our bounds are able to handle large codes, as is typically the case for a practical key generator.

$$\tilde{\mathbb{H}}_\infty(X|P) = -\log_2 \left(\sum_{\epsilon \in \mathcal{E}} \max_{\mathbf{w} \in \mathcal{W}} \mathbb{P}(X = \epsilon \oplus \mathbf{w}) \right). \tag{9}$$

Equation (8) iterates over all \mathbf{p} 's and selects each time the most likely \mathbf{x} that is within range, via the addition of a codeword $\mathbf{w} \in \mathcal{W}$. We now reverse the roles, as shown in Fig. 3. We iterate over all \mathbf{x} 's, from most likely to least likely, i.e., from φ_1 to φ_J . Within a certain φ_j , the order of the \mathbf{x} 's may be chosen arbitrarily. Subsequently, we assign \mathbf{p} 's to each \mathbf{x} , as represented by the black squares, until the set \mathcal{P} of size 2^n is depleted. For each assigned \mathbf{p} , we assume that the corresponding \mathbf{x} is the most likely vector, according to (8). Let $s_j^{\mathbf{p}}$ denote the number of black squares assigned to set φ_j . The residual min-entropy is then easily computed as in (10).

$$\tilde{\mathbb{H}}_\infty(X|P) = -\log_2 \left(\frac{1}{|\mathcal{M}|} \sum_{j=1}^J s_j^{\mathbf{p}} q_j \right). \tag{10}$$

Both linear and non-linear codes are supported by former graphical representation. Nevertheless, we elaborate linear codes as a special case due to their practical relevance. Fig. 4 swaps the order of iteration in (9). Only one row suffices, i.e., each column of helper data vectors \mathbf{p} in Figure 3 is condensed to a single square. Black and white squares are now assigned to cosets, as represented by their coset leaders ϵ . Let s_j^ϵ denote the number of black squares assigned to set φ_j . The residual min-entropy is then easily computed as in (11), hereby dropping denominator $|\mathcal{M}|$ compared to (10), given that $s_j^{\mathbf{p}} = 2^k \cdot s_j^\epsilon$.

$$\tilde{\mathbb{H}}_\infty(X|P) = -\log_2 \left(\sum_{j=1}^J s_j^\epsilon q_j \right). \tag{11}$$

In the worst-case scenario, the most likely \mathbf{x} 's all map to unique \mathbf{p} 's, without overlap, resulting in a lower bound on $\tilde{\mathbb{H}}_\infty(X|P)$. For a linear code, this would be the case if the first 2^{n-k} \mathbf{x} 's all belong to different cosets. In the best-case scenario, our sequence of \mathbf{x} 's exhibits maximum overlap in terms of \mathbf{p} , resulting in an upper bound on $\tilde{\mathbb{H}}_\infty(X|P)$. For a linear code, this would be the case if

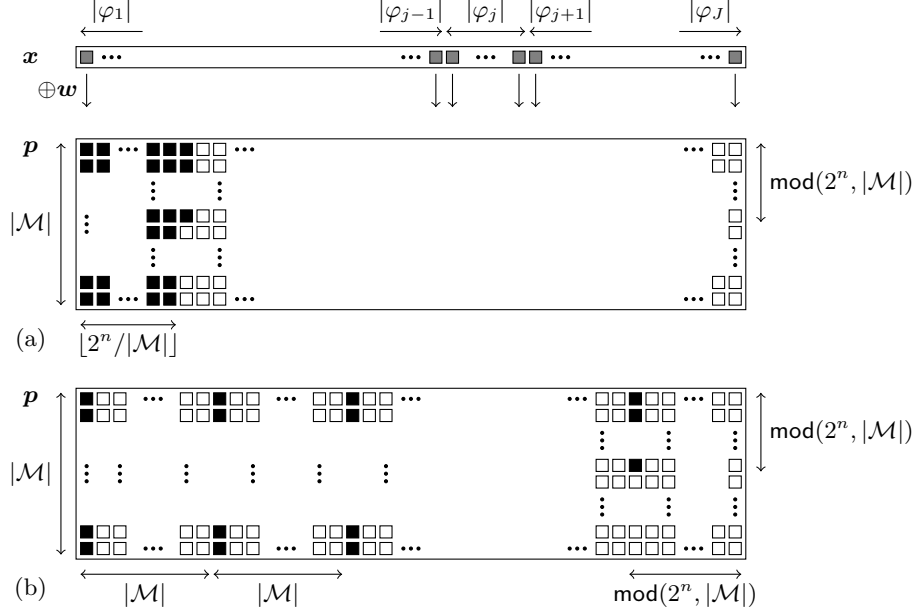


Fig. 3. Reversal of the roles in (8). (a) A lower bound on $\tilde{\mathbb{H}}_\infty(X|P)$. (b) An upper bound on $\tilde{\mathbb{H}}_\infty(X|P)$. Black squares represent terms that contribute to $\tilde{\mathbb{H}}_\infty(X|P)$, one for each $p \in \mathcal{P}$. White squares represent non-contributing terms, overruled by the max operator. In general, there are few black squares but many white squares, 2^n versus $(|\mathcal{M}| - 1)2^n$ to be precise. For block codes, i.e., $|\mathcal{M}| = 2^k$, the last column of black squares is completely filled.

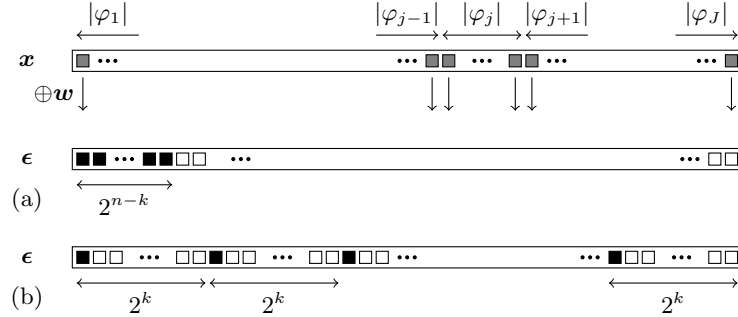


Fig. 4. Reversal of the roles in (9), as applied to linear codes. (a) A lower bound on $\mathbb{H}_\infty(X|P)$. (b) An upper bound on $\mathbb{H}_\infty(X|P)$. Black squares represent terms that contribute to $\mathbb{H}_\infty(X|P)$, one for each $\epsilon \in \mathcal{E}$. White squares represent non-contributing terms, overruled by the max operator.

the first 2^k \mathbf{x} 's all map to the same coset, and this repeated for all 2^{n-k} cosets. Algorithms 1 and 2 comprehend a literal transcript of Fig. 3 and compute the lower bound and upper bound respectively. Auxiliary variables $s^{\mathbf{p}}$ and $s^{\mathbf{x}}$ accumulate black and gray squares respectively. To maintain generality, we abstain from special case algorithms for linear codes, although it would result in a few simplifications.

Algorithm 1: BoundWorstCase	Algorithm 2: BoundBestCase
Input: List $\langle \varphi_j , q_j \rangle$ Output: Lower bound on $\tilde{\mathbb{H}}_{\infty}(X P)$ $j, q, s^{\mathbf{p}} \leftarrow 0$ while $s^{\mathbf{p}} < 2^n$ do $j \leftarrow j + 1$ $s_j^{\mathbf{p}} \leftarrow \min(\varphi_j \mathcal{M} , 2^n - s^{\mathbf{p}})$ $s^{\mathbf{p}} \leftarrow s^{\mathbf{p}} + s_j^{\mathbf{p}}$ $q \leftarrow q + s_j^{\mathbf{p}} \cdot q_j$ $\tilde{\mathbb{H}}_{\infty}(X P) \leftarrow -\log_2(q/ \mathcal{M})$	Input: List $\langle \varphi_j , q_j \rangle$ Output: Upper bound on $\tilde{\mathbb{H}}_{\infty}(X P)$ $j, q, s^{\mathbf{p}}, s^{\mathbf{x}} \leftarrow 0$ while $s^{\mathbf{p}} < 2^n$ do $j \leftarrow j + 1$ $s^{\mathbf{x}} \leftarrow s^{\mathbf{x}} + \varphi_j $ $s_j^{\mathbf{p}} \leftarrow \lceil (s^{\mathbf{x}} - s^{\mathbf{p}}) / \mathcal{M} \rceil \mathcal{M} $ $s_j^{\mathbf{p}} \leftarrow \min(\max(s_j^{\mathbf{p}}, 0), 2^n - s^{\mathbf{p}})$ $s^{\mathbf{p}} \leftarrow s^{\mathbf{p}} + s_j^{\mathbf{p}}$ $q \leftarrow q + s_j^{\mathbf{p}} \cdot q_j$ $\tilde{\mathbb{H}}_{\infty}(X P) \leftarrow -\log_2(q/ \mathcal{M})$

Algorithms 1 and 2 may now be applied to a variety of distributions. For a uniform distribution, the lower and upper bound both evaluate to $\tilde{\mathbb{H}}_{\infty}(X|P) = \log_2(|\mathcal{M}|)$, regardless of other code specifics. Or simply k , for block codes in particular. The min-entropy loss is hence exactly $(n-k)$, given that $\mathbb{H}_{\infty}(X) = n$. Reyzin's proof [24] therefore implicates that the general-purpose $(n-k)$ bound cannot be tightened any further. Although results are fairly presentable already for the biased and correlated distributions, we further tighten these bounds first.

3.3 Tighter Bounds

Tighter bounds can be obtained by leveraging code properties more effectively. Algorithms 3 and 4 generalize Algorithms 1 and 2 respectively. In the former case, an additional input imposes an upper bound on the accumulated number of black squares, i.e., $\forall j, (s_1^{\mathbf{p}} + s_2^{\mathbf{p}} + \dots + s_j^{\mathbf{p}}) \leq (u_1^{\mathbf{p}} + u_2^{\mathbf{p}} + \dots + u_j^{\mathbf{p}})$. In the latter case, an additional input imposes a lower bound on the accumulated number of black squares, i.e., $\forall j, (s_1^{\mathbf{p}} + s_2^{\mathbf{p}} + \dots + s_j^{\mathbf{p}}) \geq (l_1^{\mathbf{p}} + l_2^{\mathbf{p}} + \dots + l_j^{\mathbf{p}})$. We now provide several examples.

Worst-Case Bounds We further tighten the lower bound on $\tilde{\mathbb{H}}_{\infty}(X|P)$ for the correlated distribution. The improvement applies to linear codes that have the all-ones vector $\mathbf{1}$ of length n as a codeword. This includes Reed-Muller codes of any order [18]. This also includes many BCH, Hamming and repetition codes, on

<hr/> Algorithm 3: BoundWorstCase2 <hr/> Input: List $\langle \varphi_j , q_j, u_j^p \rangle$ Output: Lower bound on $\tilde{\mathbb{H}}_\infty(X P)$ $j, q, s^p, u^p \leftarrow 0$ while $s^p < 2^n$ do $j \leftarrow j + 1$ $u^p \leftarrow u^p + u_j^p$ $s_j^p \leftarrow \min(\varphi_j \mathcal{M} , u^p - s^p)$ $s_j^p \leftarrow \min(s_j^p, 2^n - s^p)$ $s^p \leftarrow s^p + s_j^p$ $q \leftarrow q + s_j^p \cdot q_j$ $\tilde{\mathbb{H}}_\infty(X P) \leftarrow -\log_2(q/ \mathcal{M})$ <hr/>	<hr/> Algorithm 4: BoundBestCase2 <hr/> Input: List $\langle \varphi_j , q_j, l_j^p \rangle$ Output: Upper bound on $\tilde{\mathbb{H}}_\infty(X P)$ $j, q, s^p, s^x, l^p \leftarrow 0$ while $s_{1:j}^p < 2^n$ do $j \leftarrow j + 1$ $s^x \leftarrow s^x + \varphi_j $ $l^p \leftarrow l^p + l_j^p$ $s_j^p \leftarrow \lceil (s^x - s^p) / \mathcal{M} \rceil \mathcal{M} $ $s_j^p \leftarrow \max(s_j^p, l^p - s^p, 0)$ $s_j^p \leftarrow \min(s_j^p, 2^n - s^p)$ $s^p \leftarrow s^p + s_j^p$ $q \leftarrow q + s_j^p \cdot q_j$ $\tilde{\mathbb{H}}_\infty(X P) \leftarrow -\log_2(q/ \mathcal{M})$ <hr/>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

the condition that these are cyclic and having d odd, as easily proven hereafter. Consider an arbitrary codeword with Hamming weight d . XORing all 2^n circular shifts of this codeword results in the all-ones codeword, which ends the proof. As mentioned before, each set φ_j of the correlated distribution can be partitioned in pairs $\{\mathbf{x}, \bar{\mathbf{x}}\}$, with $\bar{\mathbf{x}}$ the ones' complement of \mathbf{x} . Paired inputs belong to the same coset, i.e., maximum overlap in terms of helper data \mathbf{p} . Therefore, we impose the cumulative upper bound in (12).

$$u_j^p = |\mathcal{M}| \frac{|\varphi_j|}{2} = 2^{k-1} |\varphi_j|. \quad (12)$$

For instance, consider linear/cyclic $[n, k = 1, d = n]$ repetition codes, i.e., having generator matrix $\mathbf{G} = \mathbf{1}$, with n odd. Algorithms **BoundWorstCase2** and **BoundBestCase** then converge to the exact result $\tilde{\mathbb{H}}_\infty(X|P) = 1$, not depending on parameter c . This is the best-case scenario, given the universal bound $\tilde{\mathbb{H}}_\infty(X|P) \leq k$. Fig. 5 illustrates the former with squares for $n = 5$. The result also holds if the repetition code is neither linear/cyclic nor odd. As long as $\mathbf{w}_1 \oplus \mathbf{w}_2 = \mathbf{1}$, the elements of each φ_j can be paired into cosets. Although the term coset is usually preserved for linear codes, translations of a non-linear repetition code are either disjunct or coincide and still partition the space $\{0, 1\}^{1 \times n}$. As a side note, the result offers another [8] refutation of the *repetition code pitfall* of Koeberl et al. [17], a work that overlooks that $(n - k)$ is an upper bound only.

Best-Case Bounds We improve the upper bound on $\tilde{\mathbb{H}}_\infty(X|P)$ for both the biased and correlated distribution. In particular, we take minimum distance d into account. The main insight is that two slightly differing inputs $\mathbf{x}_u \neq \mathbf{x}_v$ do not overlap in terms of helper data \mathbf{p} . More precisely, if $\text{HD}(\mathbf{x}_u, \mathbf{x}_v) \in [1, d - 1]$, then $\{\mathbf{x}_u \oplus \mathbf{w} \mid \mathbf{w} \in \mathcal{W}\} \cap \{\mathbf{x}_v \oplus \mathbf{w} \mid \mathbf{w} \in \mathcal{W}\} = \emptyset$. For the biased distribution,

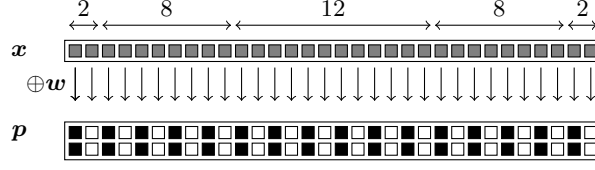


Fig. 5. The exact residual min-entropy $\tilde{\mathbb{H}}_{\infty}(X|P)$ for the correlated distribution and an $[n = 5, k = 1, d = 5]$ repetition code.

the following holds: $\text{HD}(\mathbf{x}_u, \mathbf{x}_v) \in [1, d - 1]$ if $\mathbf{x}_u \neq \mathbf{x}_v$ and $\mathbf{x}_u, \mathbf{x}_v \in (\varphi_1 \cup \varphi_2 \cup \dots \cup \varphi_{t+1})$. Or stated otherwise, the elements of the first $t + 1$ sets all result in unique \mathbf{p} 's. Therefore, we can impose the constraint given in (13). Fig. 6 depicts the squares.

$$l_j^{\mathbf{p}} = \begin{cases} |\varphi_j| |\mathcal{M}|, & \text{if } j \in [1, t + 1] \\ 0, & \text{otherwise} \end{cases}. \quad (13)$$

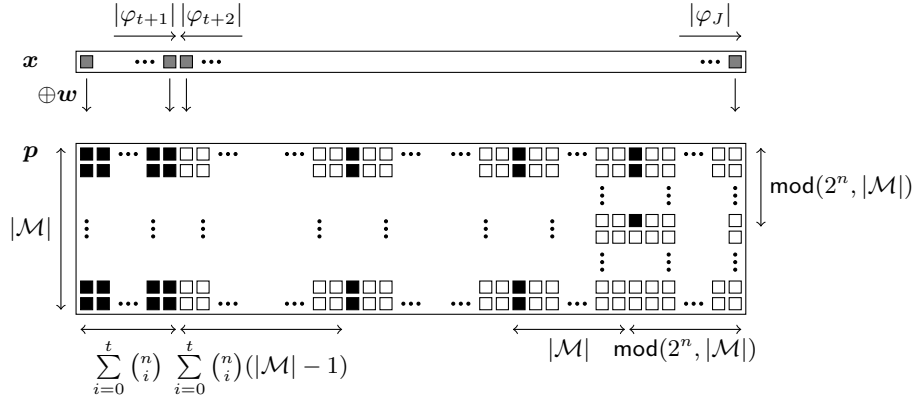


Fig. 6. A tightened upper bound on $\tilde{\mathbb{H}}_{\infty}(X|P)$ for the biased distribution, hereby making use of (13).

There is an interesting observation for perfect codes in particular. As clear from the Hamming bound in (3), all unique \mathbf{p} 's are covered by the first $t + 1$ sets exclusively. `BoundWorstCase` and `BoundBestCase2` hence produce the same output, implying that the residual min-entropy is evaluated exactly, as further simplified in (14). Delvaux et al. [8] derived the same formula for $[n, k = 1, d = n]$ repetition codes with n odd. The scope of their result is hence extended from perfect repetition codes to perfect codes in general. As a side note, the formula was originally adopted to debunk the aforementioned *repetition code pitfall* [17].

Maes et al. [22] later presented a similar contribution at CHES 2015, differing in its use of Shannon entropy rather than min-entropy.

$$\tilde{\mathbb{H}}_{\infty}(X|P) = -\log_2\left(\sum_{j=1}^{t+1} |\varphi_j| \cdot q_j\right) = -\log_2(\text{F}_{\text{bino}}(t; n, \min(b, 1-b))). \quad (14)$$

Also for the correlated distribution, distance d might be incorporated to tighten the upper bound on $\tilde{\mathbb{H}}_{\infty}(X|P)$. First of all, we assign $|\mathcal{M}|$ unique \mathbf{p} 's to one out of two elements in φ_1 . For ease of understanding, assume $\mathbf{x} = \mathbf{0}$, comprehending the first case in (15). For each set φ_j , with $j \in [2, n]$, we then count the number of inputs $\mathbf{x} \in \varphi_j$ such that $h = \text{HW}(\mathbf{x}) \leq t$. The latter constraint guarantees all assigned \mathbf{p} 's to be unique. We distinguish between two forms, $\mathbf{x} = (\mathbf{0} \parallel \mathbf{1} \parallel \mathbf{0} \parallel \dots)$ and $\mathbf{x} = (\mathbf{1} \parallel \mathbf{0} \parallel \mathbf{1} \parallel \dots)$, resulting in two main terms. For each form, we apply *stars and bars* combinatorics twice. In particular, we assign h indistinguishable stars, i.e., ones, to distinguishable bins and independently also for $n - h$ zeros. Note that $l_j^{\mathbf{p}} = 0$ for $j > 2t + 1$. To ensure formula correctness, one may verify numerically that $l_1^{\mathbf{p}} + l_2^{\mathbf{p}} + \dots + l_{2t+1}^{\mathbf{p}}$ equals the left hand side of the Hamming bound in (3).

$$l_j^{\mathbf{p}} = \begin{cases} |\mathcal{M}|, & \text{if } j = 1 \\ |\mathcal{M}| \left(\sum_{h=\lfloor j/2 \rfloor}^t \binom{h-1}{\lfloor j/2 \rfloor - 1} \binom{n-h-1}{\lfloor j/2 \rfloor - 1} \right. \\ \quad \left. + \sum_{h=\lceil j/2 \rceil}^t \binom{h-1}{\lceil j/2 \rceil - 1} \binom{n-h-1}{\lceil j/2 \rceil - 1} \right), & \text{otherwise.} \end{cases} \quad (15)$$

3.4 Numerical Results

Fig. 7 presents numerical results for various BCH codes. We focus on small codes, as these allow for an exact exhaustive evaluation of the residual min-entropy using (8) and/or (9). As such, the tightness of various bounds can be assessed adequately. Fig. 7(d) nevertheless demonstrates that our algorithms support large codes equally well, in compliance with a practical key generator. Note that only half of the bias interval $b \in [0, 1]$ is depicted. The reason is that all curves mirror around the vertical axis of symmetry $b = \frac{1}{2}$. The same holds for the correlated distribution with parameter c .

Especially the lower bounds perform well, which benefits a conservative system provider. The best lower bounds in Figs. 7(a), (b) and (c) visually coincide with the exact result. The gap with the $(n - k)$ bound is the most compelling around $b, c \approx 0.7$, where the corresponding curves hit the horizontal axis $\mathbb{H}_{\infty}(X|P) = 0$. Also our upper bounds are considerably tighter than their more general alternatives in (5). Nevertheless, the latter bounds remain open for further improvement, with the exception of Fig. 7(b). An $[n = 7, k = 4, d = 3]$ code is perfect and lower and upper bounds then converge to the exact result for a biased distribution.

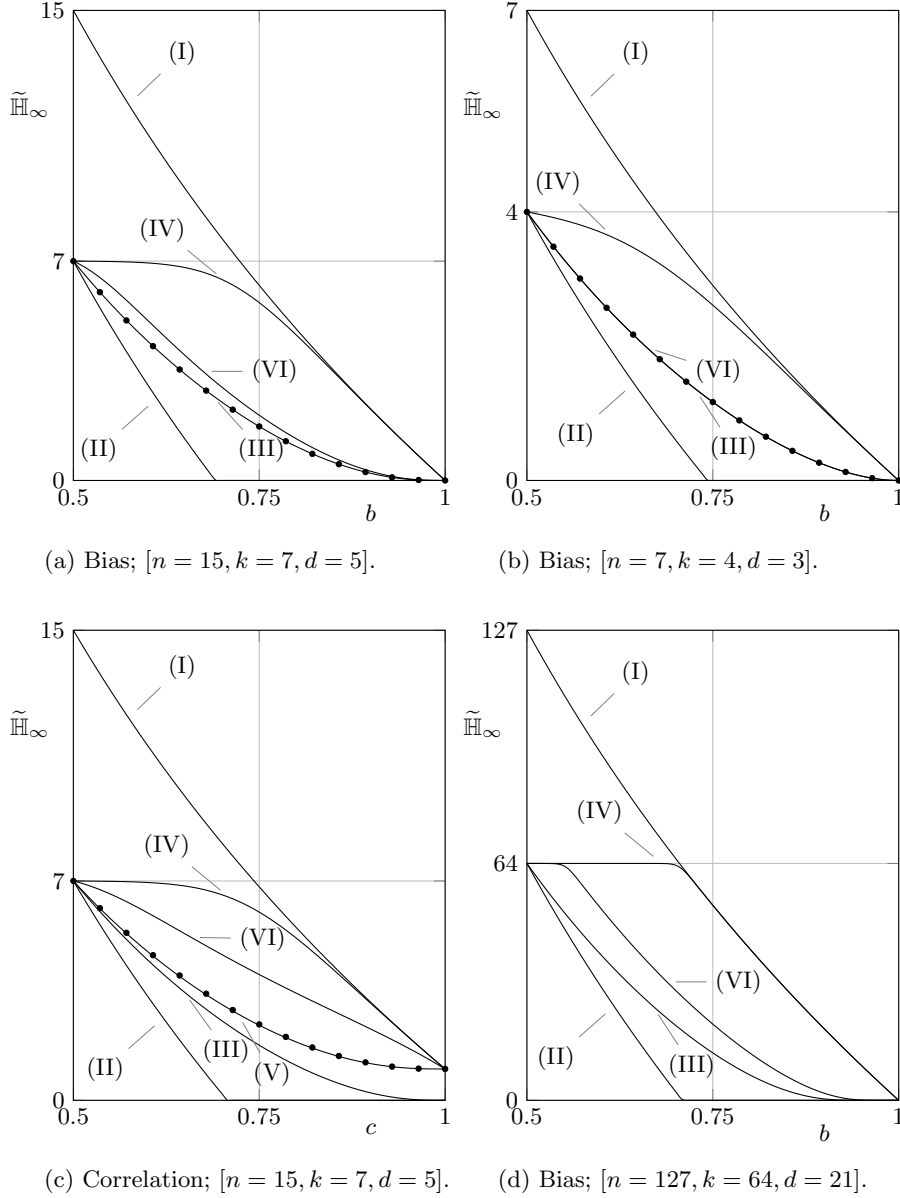


Fig. 7. The secure sketch min-entropy loss for various BCH codes. Dots correspond to an exact exhaustive evaluation of (8)/(9). The legend of the curves is as follows. (I) The ingoing min-entropy $\mathbb{H}_\infty(X) = -\log_2(q_1)$. (II) The lower bound $\tilde{\mathbb{H}}_\infty(X|P) = \max(\mathbb{H}_\infty(X) - (n - k), 0)$. (III) The lower bound on $\tilde{\mathbb{H}}_\infty(X|P)$ according to **BoundWorstCase**. (IV) The upper bound on $\tilde{\mathbb{H}}_\infty(X|P)$ according to **BoundBestCase**. (V) The lower bound on $\tilde{\mathbb{H}}_\infty(X|P)$ according to **BoundWorstCase2**. (VI) The upper bound on $\tilde{\mathbb{H}}_\infty(X|P)$ according to **BoundBestCase2**.

4 Applications

The newly developed theory of Section 3 facilitates the design and analysis of error-correction methods for PUFs, as exemplified in twofold manner. First, we point out a fundamental security flaw in the reverse fuzzy extractor [28]. Second, we provide a motivational framework for debiasing schemes [30, 15, 26, 27, 22].

4.1 A Fundamental Security Flaw in Reverse Fuzzy Extractors

The reverse fuzzy extractor, as proposed by Van Herrewege et al. [28] at Financial Crypto 2012, improves the lightweight perspectives of PUF-based authentication protocols. The construction was therefore also adopted in the CHES 2015 protocol of Aysu et al. [1]. Instead of a single helper data exposure only, $\mathbf{p} \leftarrow \text{SSGen}(\tilde{\mathbf{x}})$ is regenerated and transferred with each protocol run by a resource-constrained PUF-enabled device. A receiving resource-rich server, storing reference response \mathbf{x} , can hence reconstruct $\tilde{\mathbf{x}} \leftarrow \text{SSRec}(\mathbf{x}, \mathbf{p})$ and establish a shared secret as such. The footprint of the device is reduced due to the absence of the heavyweight SSRec procedure.

We debunk the main security claim that repeated helper data exposure does not result in additional min-entropy loss. The revealed flaw is attributed to the misuse of a reusability proof of Boyen [6]. For the code-offset sketch with linear codes, the exposure of $\mathbf{p}_1 \leftarrow \text{SSGen}(\mathbf{x})$ and $\mathbf{p}_2 \leftarrow \text{SSGen}(\mathbf{x} \oplus \mathbf{e})$, with perturbation \mathbf{e} known and fully determined by the attacker, is provably equivalent. The latter helper data reveals that \mathbf{x} belongs to an identical coset $\{\mathbf{p}_1 \oplus \mathbf{w} : \mathbf{w} \in \mathcal{W}\} = \{\mathbf{p}_2 \oplus \mathbf{e} \oplus \mathbf{w} : \mathbf{w} \in \mathcal{W}\}$. However, perturbation \mathbf{e} is determined by PUF noisiness rather than by the attacker and its release hence reveals new information. Given a sequence of protocol runs, the attacker can approximate all individual bit error rates p_E as well as the coset to which reference \mathbf{x} belongs.

Fig. 8 quantifies the residual min-entropy of X with the exclusion and inclusion of revealed bit error rates p_E respectively. In the latter case, we rely on a Monte Carlo evaluation of (16), as enabled by choosing a small $[n = 15, k = 7, d = 5]$ BCH code, given that an analytical approach is not so very straightforward. Exposure of p_E boils down to knowledge of threshold discrepancy $|\mathbf{v}(i) - t|$. For the biased distribution, the situation is identical to the flaw in the *soft-decision decoding* scheme of Maes et al. [21]. As pointed out by Delvaux et al. [8], there is a bit-specific bias $b_i = \mathbb{P}(\mathbf{r}(i) = 1) = f_{\text{norm}}(t + |\mathbf{v}(i) - t|) / (f_{\text{norm}}(t + |\mathbf{v}(i) - t|) + f_{\text{norm}}(t - |\mathbf{v}(i) - t|))$. For each \mathbf{x} in the coset corresponding to \mathbf{p} , we then compute $\mathbb{P}(X = \mathbf{x}) = \prod_{i=1}^n (\mathbf{x}(i)b_i + (1 - \mathbf{x}(i))(1 - b_i))$. Similarly, for the spatially correlated distribution, we compute $\mathbb{P}(X = \mathbf{x}) = f_{\text{norm}}(\mathbf{v}, \mathbf{0}, \mathbf{\Sigma})$, with covariance matrix $\mathbf{\Sigma}$ exclusively depending on correlation parameter c , as detailed in the extended version of this manuscript.

$$\tilde{\mathbb{H}}_{\infty}(X|P) = -\log_2 \left(\mathbb{E}_{\mathbf{v} \leftarrow V} \max_{\mathbf{w} \in W} \mathbb{P}(V = t + (1 - 2\mathbf{w})|\mathbf{v} - t| \mid |\mathbf{v} - t|) \right). \quad (16)$$

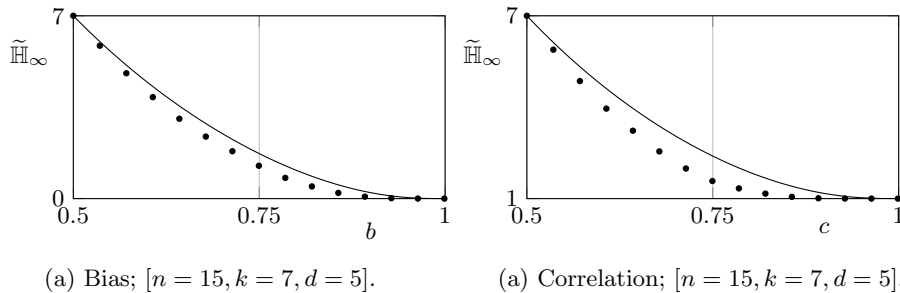


Fig. 8. The residual min-entropy $\hat{\mathbb{H}}_\infty(X|P)$ for a BCH code. The solid lines that exclude revealed bit error rates are computed with `BoundWorstCase2`; Fig. 7 confirms the visual overlap with the exact result. Dots that include revealed bit error rates correspond to Monte Carlo evaluations of size 10^6 .

The revealed flaw differs from existing attacks by Delvaux et al. [9] and Becker [3] that apply to the original protocol [28] exclusively. The latter attacks comprehend the modeling of the highly correlated arbiter PUF via repeated helper data exposure; a preemptive fix can be found in the PhD thesis of Maes [19]. The newly revealed flaw is more fundamentally linked to the reverse fuzzy extractor primitive and applies to all existing protocols so far [28, 19, 1]. Observe in Fig. 8 that the overly conservative $(n-k)$ bound would compensate for the additional unanticipated min-entropy loss. However, this somewhat defeats the purpose in light of the original lightweight intentions, and this observation might not necessarily hold for every possible distribution. Further theoretical work may determine to which extent and at which cost reverse fuzzy extractors can be repaired. A potential fix already exists for biased distributions, as illustrated later-on.

4.2 Motivation for Debiasing Schemes

Debiasing schemes transform a biased PUF-induced distribution into a uniform distribution. A considerable fraction of the response bits is discarded in order to restore the balance between 0 and 1. Indices of retained bits are stored as helper data. A subsequent secure sketch, known to have an exact min-entropy loss of $(n-k)$ bits for uniform inputs, still corrects the errors. A first debiasing proposal is the *index-based syndrome* (IBS) scheme of Yu et al. [30], further generalized by Hiller et al. [15]. Second, several variations of the von Neumann debiasing algorithm can be applied. This was first proposed by van der Leest et al. [26], and later also by Van Herrewege in his PhD thesis [27]. Most recently, Maes et al. [22] presented an optimization of the von Neumann algorithm that applies to repetition codes in particular.

Prior debiasing proposals conjectured that a stand-alone sketch cannot handle biased distributions well. This conclusion originates from the extrapolation of repetition code insights and/or application of the $(n-k)$ bound. The precise

entropy loss behavior for larger codes, e.g., a BCH $[n = 127, k = 64, d = 21]$ code as in Fig. 7, was an educated guess so far. Our newly derived bounds clearly resolve this motivational uncertainty, in addition to making stand-alone sketches more competitive. For low-bias situations, the $(n - k)$ bound already resulted in a competitive sketch [22]; the new bounds can only improve hereupon. We emphasize that modern high-quality PUFs tend to have a low bias. Notable cases of a high bias can typically be attributed to an avoidable asymmetry in the circuit. Nevertheless, for high-bias situations, the new bounds clearly indicate the need of debiasing schemes. The benefit is amplified by choosing a sketch with a k -bit output, several of which are listed in the extended version of this manuscript. The uniform output is then directly usable as a key, hereby eliminating the Hash function and its additional min-entropy loss in case the leftover hash lemma is applied.

Finally, we highlight that one of the von Neumann debiasing schemes in [22] was claimed to be reusable. This claim holds, despite overlooking the misuse of Boyen’s proof and stating that a stand-alone sketch is reusable. A side effect of retaining pairs of alternating bits only, i.e., 01 and 10, is that the imbalance in error rates between 0 and 1 cannot be observed in the helper data. The scheme is considerably less efficient than other von Neumann variants though, showing that reusability comes at a price.

5 Conclusion

Secure sketches are the main workhorse of modern PUF-based key generators. The min-entropy loss of most sketches is upper-bounded by $(n - k)$ bits and designers typically instantiate system parameters accordingly. However, the latter bound tends to be overly pessimistic, resulting in an unfortunate implementation overhead. We showcased the proportions for a prominent category of PUFs, with bias and spatial correlations acting as the main non-uniformities. New considerably tighter bounds were derived, valid for a variety of popular but algebraically complex codes. These bounds are unified in the sense of being applicable to seven secure sketch constructions. Deriving tighter alternatives for the $(n - k)$ bound counts as unexplored territory and we established the first significant stepping stone. New techniques may have to be developed in order to tackle more advanced *second-order* distributions. Elaborating a wider range of applications would be another area of progress. We hope to have showcased the potential by debunking the main security claim of the reverse fuzzy extractor and by providing proper quantitative motivation for debiasing schemes.

Acknowledgment

The authors greatly appreciate the support received. The European Union’s Horizon 2020 research and innovation programme under grant number 644052 (HECTOR). The Research Council of KU Leuven, GOA TENSE (GOA/11/007),

the Flemish Government through FWO G.0550.12N and the Hercules Foundation AKUL/11/19. The national major development program for fundamental research of China (973 Plan) under grant number 2013CB338004. Jeroen Delvaux is funded by IWT-Flanders grant number SBO 121552. Matthias Hiller is funded by the German Federal Ministry of Education and Research (BMBF) in the project SIBASE through grant number 01IS13020A.

References

1. A. Aysu, E. Gulcan, D. Moriyama, P. Schaumont, and M. Yung. End-To-End Design of a PUF-Based Privacy Preserving Authentication Protocol. In *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, September 13-16, 2015, Proceedings*, pages 556–576, 2015.
2. B. Barak, Y. Dodis, H. Krawczyk, O. Pereira, K. Pietrzak, F. Standaert, and Y. Yu. Leftover Hash Lemma, Revisited. In *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference*, pages 1–20, 2011.
3. G. T. Becker. On the Pitfalls of Using Arbiter-PUFs as Building Blocks. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 34(8):1295–1307, 2015.
4. M. Bhargava and K. Mai. An efficient reliable PUF-based cryptographic key generator in 65nm CMOS. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2014, Dresden, Germany, March 24-28, 2014*, pages 1–6, 2014.
5. C. Bösch, J. Guajardo, A. Sadeghi, J. Shokrollahi, and P. Tuyls. Efficient Helper Data Key Extractor on FPGAs. In *Cryptographic Hardware and Embedded Systems - CHES 2008, 10th International Workshop*, pages 181–197, 2008.
6. X. Boyen. Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 82–91, 2004.
7. L. Carter and M. N. Wegman. Universal Classes of Hash Functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.
8. J. Delvaux, D. Gu, D. Schellekens, and I. Verbauwhede. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 2015.
9. J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede. A Survey on Lightweight Entity Authentication with Strong PUFs. *ACM Comput. Surv.*, 48(2):26, 2015.
10. J. Delvaux and I. Verbauwhede. Fault Injection Modeling Attacks on 65 nm Arbiter and RO Sum PUFs via Environmental Changes. *IEEE Trans. on Circuits and Systems*, 61-I(6):1701–1713, 2014.
11. Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing*, 38(1):97–139, 2008.
12. W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. 1968.
13. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A Pseudorandom Generator from any One-way Function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.
14. A. V. Herwege, V. van der Leest, A. Schaller, S. Katzenbeisser, and I. Verbauwhede. Secure PRNG seeding on commercial off-the-shelf microcontrollers. In *TrustED’13, Proceedings of the 2013 ACM Workshop on Trustworthy Embedded Devices*, pages 55–64, 2013.

15. M. Hiller, D. Merli, F. Stumpf, and G. Sigl. Complementary IBS: application specific error correction for PUFs. In *2012 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2012, June 3-4, 2012*, pages 1–6, 2012.
16. D. E. Holcomb, W. P. Burleson, and K. Fu. Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers*, 58(9):1198–1210, 2009.
17. P. Koeberl, J. Li, A. Rajan, and W. Wu. Entropy loss in PUF-based key generation schemes: The repetition code pitfall. In *2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014, Arlington, VA, USA, May 6-7, 2014*, pages 44–49, 2014.
18. F. J. MacWilliams and N. J. A. Sloane. *The theory of error correcting codes*. 1977.
19. R. Maes. *Physically Unclonable Functions: Constructions, Properties and Applications*. PhD thesis, KU Leuven, 2012. Ingrid Verbauwhede (promotor).
20. R. Maes. An accurate probabilistic reliability model for silicon PUFs. In *Cryptographic Hardware and Embedded Systems - CHES 2013 - 15th International Workshop, Santa Barbara, CA, USA, August 20-23, 2013. Proceedings*, pages 73–89, 2013.
21. R. Maes, P. Tuyls, and I. Verbauwhede. A Soft Decision Helper Data Algorithm for SRAM PUFs. In *ISIT 2009, IEEE International Symposium on Information Theory*, pages 2101–2105, 2009.
22. R. Maes, V. van der Leest, E. van der Sluis, and F. Willems. Secure key generation from biased PUFs: extended version. *Journal of Cryptographic Engineering*, 6(2):121–137, 2016.
23. R. Maes, A. Van Herrewege, and I. Verbauwhede. PUFKY: A Fully Functional PUF-Based Cryptographic Key Generator. In *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop*, pages 302–319, 2012.
24. L. Reyzin. Entropy Loss is Maximal for Uniform Inputs. Technical Report BUCS-TR-2007-011, Department of Computer Science, Boston University, September 2007.
25. P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters. Read-Proof Hardware from Protective Coatings. In *CHES 2006, Int. Workshop on Cryptographic Hardware and Embedded Systems*, pages 369–383, 2006.
26. V. van der Leest, G.-J. Schrijen, H. Handschuh, and P. Tuyls. Hardware Intrinsic Security from D Flip-flops. In *Proceedings of the Fifth ACM Workshop on Scalable Trusted Computing, STC '10*, pages 53–62, 2010.
27. A. Van Herrewege. *Lightweight PUF-based Key and Random Number Generation*. PhD thesis, KU Leuven, 2015. Ingrid Verbauwhede (promotor).
28. A. Van Herrewege, S. Katzenbeisser, R. Maes, R. Peeters, A. Sadeghi, I. Verbauwhede, and C. Wachsmann. Reverse Fuzzy Extractors: Enabling Lightweight Mutual Authentication for PUF-Enabled RFIDs. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, February 27-March 2, 2012, Revised Selected Papers*, pages 374–389, 2012.
29. H. Yu, P. H. W. Leong, H. Hinkelmann, L. Möller, M. Glesner, and P. Zipf. Towards a Unique FPGA-Based Identification Circuit Using Process Variations. In *FPL 2009, Int. Conference on Field Programmable Logic and Applications*, pages 397–402, 2009.
30. M. Yu and S. Devadas. Secure and Robust Error Correction for Physical Unclonable Functions. *IEEE Design & Test of Computers*, 27(1):48–65, 2010.